

WHAT IS CLAIMED IS:

1. A system for synchronizing a plurality of replicated databases at least intermittently communicating with one another, comprising:

a local replicated database;

5 an interface for communicating with one or more remote replicated database via a communications link; and

a synchronization manager associated with the local replicated database for sending changes made on the local replicated database to one or more remote replicated databases for reconstruction by the one or more remote replicated databases, receiving changes made on a remote replicated database, and reconstructing changes received from a remote replicated database on the local replicated database.

2. The system of claim 1, wherein the synchronization manager is configured for sending changes, receiving changes, and reconstructing changes independently from one another.

3. The system of claim 1, further comprising a sequence table associated with the local replicated database for tracking changes sent to and received from remote replicated databases.

4. The system of claim 3, wherein the synchronization manager is configured for sending changes to one ore more remote replicated databases in one or more change files, and wherein each change file comprises a local sequence number identifying the local remote

database and a remote sequence number identifying each remote database to which the change file was sent, the local and remote sequence numbers being stored in the sequence table.

5        5.        The system of claim 3, wherein the synchronization manager is configured for receiving changes from one or more remote replica databases in one or more change files, and wherein each change file comprises a local sequence number identifying the local remote database, and a remote sequence number identifying the remote database from which each change file was sent, the local and remote sequence numbers being stored in the sequence table.

10       6.       The system of claim 1, wherein the synchronization manager is configured for monitoring activity of the local replicated database, and for reconstructing changes in a manner that substantially minimizes interference with operation of the local replicated database.

15       7.       A system for synchronizing a plurality of replicated databases at least intermittently communicating with one another, comprising:  
         a first replicated database;  
         a second replicated database at least intermittently disconnected from the first replicated database;  
         at least one of the first and second replicated databases comprising an interface for  
20       communicating with each other via a communications link;  
         a first synchronization manager associated with the first replicated database for sending changes made on the first replicated database to the second replicated database, receiving

changes made on the second replicated database, and reconstructing changes received from the second replicated database on the first replicated database; and

a second synchronization manager associated with the second replicated database for sending changes made on the second replicated database to the first replicated database,  
5 receiving changes made on the first replicated database, and reconstructing changes received from the first replicated database on the second replicated database.

8. The system of claim 7, wherein the first and second synchronization managers are configured for reconstructing changes autonomously from one another.

9. The system of claim 7, further comprising a sequence table associated with the first replicated database for tracking changes sent to and received from the second replicate database.

10. The system of claim 9, wherein the synchronization manager is configured for receiving changes from one or more remote replica databases in one or more change files, and wherein each change file comprises a local sequence number identifying the local remote database, and a remote sequence number identifying the remote database from which each change file was sent, the local and remote sequence numbers being stored in the sequence table.

11. A method for synchronizing a local replicated database with one or more remote replicated databases, comprising:

sending recent local changes made on the local database to a remote database;

receiving changes made on the remote database from the remote database; and  
reconstructing the received changes received from the remote database on the local  
database,

wherein any of the sending, receiving, and reconstructing steps may be performed  
5 independently from each other.

12. The method of claim 11, wherein the local and remote databases are version-  
managed databases, each having a plurality of versions, and wherein one version in the local  
database is nominated as an interface version, and wherein the reconstructing step is performed  
10 using the interface version.

13. The method of claim 12, wherein changes sent between the local and remote  
databases are sent in change files, and wherein the reconstructing step comprises:

(a) creating a child version from the interface version;

15 (b) setting the child version to a state identical to a state at the remote database before a  
received change file comprising the received changes were made at the remote database;

(c) loading the changes in the received change file into the child version;

(d) determining whether there has been more than one change file sent by the local  
database since a most-recent change file sent by the local database and received and processed  
20 by the remote database before the remote database sent the received change file;

(e) if there has been more than one change sent by the local database, reconciling the  
interface version with any intervening states of the local database following the state identified  
by the change file received from the remote database and preceding the interface version;

- (f) reconciling the interface version with the child version; and
- (g) posting the state comprising the results of reconciling the child version with the interface version to the interface version.

5           14.     The method of claim 13, wherein the step of reconciling the interface version with any intervening states comprises:

- (i) creating a grandchild version from the interface version;
- (ii) setting the grandchild version to a state represented in a change file sent by the local database sequentially after the most-recent change file, and henceforth taking a “current” change file to be that sequential change file;

- (iii) reconciling the grandchild version with the child version;
- (iv) posting the state comprising results of reconciling the grandchild version with the child version to the child version; and

- (v) if another change file was sent from the local database after the “current” change file in the state to which the grandchild version in step (f) was pointing but before the state to which the interface version is pointing, then repeating steps (ii) through (v) using each change file sent by the local database following sequentially after the most-recent change file.

15           15.     The method of claim 14, wherein the states are implicit.

20           16.     The method of claim 14, wherein the states are explicit.

17. The method of claim 14, wherein the step of reconciling the grandchild version with the child version, comprises:

merging differences between the grandchild and child versions; and

resolving any conflicts between the grandchild and child versions according to a set of

5 preset rules.

18. The method of claim 17, wherein the step of reconciling the grandchild version with the child version, further comprises:

creating a first park version and a second park version; and

10 setting the grandchild version in the first park version and setting the child version in the second park version if data in the grandchild version conflicts with data in the child version.

19. The method of claim 14, wherein the step of reconciling the interface version with the child version, comprises:

15 merging differences between the interface and child versions; and

resolving any conflicts between the interface and child versions according to a set of preset rules.

20. The method of claim 19, wherein the step of reconciling the interface version with the child version, further comprises:

creating a first park version and a second park version; and

20 setting the interface version in the first park version and setting the child version in the second park version if data in the interface version conflicts with data in the child version.

21. A method for synchronizing a local replicated database with a remote replicated database, comprising the steps of:

autonomously and asynchronously sending changes made on the local database to the remote database, independent of any steps of receiving and reconstructing changes;

autonomously and asynchronously receiving changes made on the remote database to the local database, independent of any steps of sending and reconstructing changes; and

autonomously and asynchronously reconstructing received changes made on the remote database on the local database, independent of any steps of sending and receiving changes.

22. The method of claim 21, wherein each of the local and remote databases is a version-managed database comprising a plurality of versions, and wherein a version in each of the local and remote databases is nominated as a local interface version and a remote interface version, respectively.

23. The method of claim 22, wherein each of the local and remote databases comprises a plurality of states and a sequence table comprising sequence numbers for identifying respective states.

24. The method of claim 22, wherein the states are explicit.

25. The method of claim 22, wherein the states are implicit.

26. The method of claim 22, wherein changes sent between the local and remote databases are sent in change files.

27. The method of claim 26, wherein each change file sent by the local database comprises a local sequence number identifying a state of the local database at the time the change file is sent and a remote sequence number identifying a state of the remote database known by the local database, and wherein each change file sent by the remote database comprises a remote sequence number identifying a state of the remote database at the time the change file is sent and a local sequence number identifying a state of the local database known by the remote database at the time the change file is sent.

28. The method of claim 27, wherein the reconstructing step comprises:

(a) creating a child version of the local database set to a state associated with a local sequence number in a received change file received from the remote database;

(b) loading the received change file into the child version;

(c) determining whether a difference between a sequence number associated with the local interface version and the local sequence number of the change file is more than one;

(d) if the difference is more than one, reconciling intervening versions of the local database, comprising:

(i) creating a grandchild version of the local database set to a state associated with a sequence number immediately following the local sequence number of the received change file;

(ii) reconciling the child version with the grandchild version;



(iii) posting reconciliation results of reconciling the child version with the grandchild version to the child version;

(iv) if a difference between the local sequence number of the local interface version and the sequence number of the child version is more than one, then repeating steps (i) to (iv) using a state of the local database associated with a sequence number next following the local sequence number of the received change file;

(e) reconciling the interface version with the child version; and

(f) posting a state comprising reconciliation results of reconciling the interface version with the child version to the interface version.

29. The method of claim 28, wherein the step of reconciling the interface version with the child, comprises the steps of:

merging differences between the interface and child versions; and

resolving any conflicts between the interface and child versions according to a preset

rules.

30. The method of claim 29, wherein the step of reconciling the interface version with the child version, further comprises the step of:

creating a first park version and a second park version; and

setting the interface version in the first park version and setting the child version in the second park version if data in the interface version conflicts with data in the child version.